A Shape-Free, Designable 6-DoF Marker Tracking Method for Camera-Based Interaction in Mobile Environment

Hiroki Nishino
Graduate School for Integrative Sciences & Engineering,
National University of Singapore
CeLS, #05-01, 27 Med. Drive Singapore 117456
+(65)-6516-1480
g0901876@nus.edu.sg

ABSTRACT
We developed a novel marker tracking method with shape-free, designable markers, which can be visually meaningful to users. The method can work fast enough to provide a real-time camera-based interaction even on low performance CPUs such as ones used in mobile Internet devices. Features such as visually communicative design and inexpensive computational cost are very desirable for users with mobile devices in the mobile/pervasive interaction environment.

The method utilizes the topological region adjacency to detect the marker candidates and then apply a simple method similar to geometric-hashing to determine the detected maker by voting to the hash tables. By such a combination of two different approaches, our method can distinguish those markers with the same topological structure and is also capable of 6-DoF pose estimation whereas most of the existing topology-based systems can not distinguish markers with the same topological structure and are incapable of 6-DoF pose estimation.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – Input devices and strategies.

General Terms
Algorithms, Design, Experimentation, Human Factors

Keywords
Visual marker recognition, visual marker design, mobile devices, mobile HCI, fiducial recognition, augmented reality

1. INTRODUCTION
Camera-based interaction is one of the most popular methods to build an intuitive and easily accessible interactive system for mobile devices. While markerless object recognition systems that involve SIFT/SURF-based techniques [8,1] are recently gaining popularity, they are still too computationally expensive to run in real-time on a cheap mobile device with a low performance CPU. Markers are still widely used for camera-based interaction on such mobile devices.

MM'10, October 25–29, 2010, Firenze, Italy.

Typically such markers are in square shapes and filled with matrix patterns that encode binary data such as ones found in QRCode [6], ARTag [5], ARToolkit Plus[11], Cyber Code [10] and the like. ARToolkit [7] uses a visually meaningful pattern, but the designability is significantly limited by its thick square-shaped frame. Lack of such visually communicative design can be a significant obstacle especially in the mobile/pervasive environment, where a user must first notice a marker in their present location and understand its meaning before initiating any interaction, unlike in an immersive environment where a user always wear an head-mounted-display, which continuously displays information on the detected markers.

In such mobile/pervasive environments, it is desirable to provide a visual cue to the users by its own appearance, to imply the kind of information associated with a marker; however, most of the existing marker-tracking methods fail to provide such a visual cue to users, due to the limitation in marker design and shape.

Costanza and his colleagues developed a marker tracking system called *d-touch* [3], with a considerable focus on such a visually communicative design issue [3, 4]. Yet, due to the limitation of its tracking method that utilizes only topological information, *d-touch* can not distinguish those markers with different appearances if their topological structures are identical. It is also incapable of 6-DoF pose estimation.

We present a novel marker tracking method, extending the topology-based marker tracking technique by combining a method similar to geometric-hashing. Such a hybrid approach for marker recognition makes it possible to distinguish the different markers with the same topological structure and acquire 6-DoF pose information. Lack of these features has been a major deficit of most of the existing topology-based marker tracking systems.

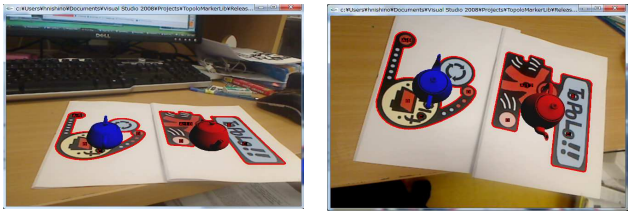

Figure 1. Pictures from our prototype system

Using the topological information as additional information also in the phase when we apply a geometry-based method, the additional computational cost for marker detection in our method is considerably reduced. Our prototype system can work in real-time even on a mobile Internet device with a low performance CPU. Figure 1 above shows some example pictures from our prototype systems. Two different markers with the same

topological structures are correctly distinguished . 3D models are projected over each marker, using 6DoF pose estimation.

2. RELATED WORKS

In this section, to clarify our contribution, we first briefly review two major marker tracking systems, *d-touch* and *reacTIVision* [2], both of which are based on topology-based approach. Then we describe geometric-hashing method very briefly since we use almost the same kind of the strategy to use votes to a hash table to detect a certain model in a given input.

2.1 D-touch

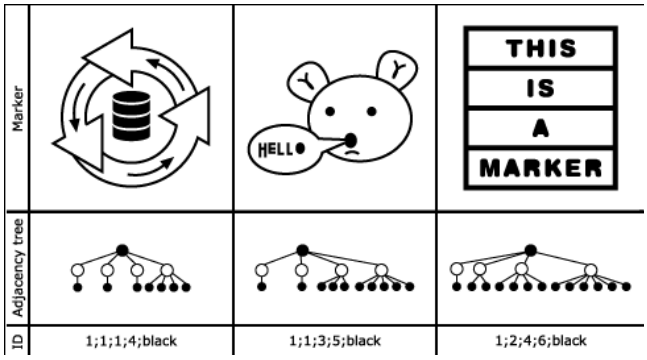

Figure 2. Examples of *d-touch* markers and their topological region adjacency trees [3]

Figure 2 above shows the examples of the *d-touch* markers, taken from [3]. As shown by d-touch markers in Figure 2, the topology-based method allows a visually expressive design and can supply meaningful implication to users. Such a feature is derived from its topology-based marker tracking method.

A binarized image can be interpreted as a tree of region adjacency, or containership information of black/white regions. Each marker in Figure 2 can be interpreted as the tree structure below it. To detect a marker, *d-touch* seeks for the same tree structure of topological region adjacency from the given input. Since the shape of each region in a marker can be altered freely as long as the topological structure stays the same, such a topology-based method leaves considerable freedom in the visual design of markers.

However, such a freedom from geometry in this topology-based method is also a source of its deficits. Because of the lack of geometric information, it can not distinguish those markers with the same topological structures, even when they have totally different appearances to human eyes. Furthermore, since 6-DoF pose estimation requires at least the 4 correspondent points between given input image and a marker model, it is impossible to achieve by such methods based solely on topological information.

2.2 ReacTIVision

The method of finding the topological structures of pre-registered markers from given input image is also an important issue when designing a topology-based marker tracking system. Kaltenbrunner and his colleagues developed a topology-based marker tracking system called *reacTIVision*, which uses left heavy depth sequence [9] to find the topological structures of fiducial markers from a input image. Left heavy depth sequence is a canonical sequence that can describe the structures of unordered rooted trees. *ReacTIVision* uses this left heavy depth sequence as a string to map a detected marker candidate to its own unique ID.

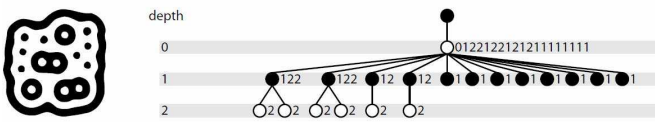

Figure 3. An example of *reacTIVision* fiducial marker and its left heavy depth sequence [2].

Figure 3 above is an example of a *reacTIVision* marker and its region adjacency tree, which appears in their paper [2]. The tree is sorted in left heavy depth order. Its left heavy depth sequence of the tree is 012212212112111111 as described.

2.3 Geometric-Hashing

Geometric hashing is one of the widely used techniques in computer vision, which is comprised of the preprocessing phase to register models to a hash table and the recognition phase to vote the various points of interest to the hash table to recognize a model in the given input. Wolfson's article on geometric-hashing [12] describes the method in the detail. Figure 4 below taken from the article briefly explains the method.

Preprocessing phase

For each model *m* do the following:

1. Extract the model's point features. Assume that *n* such features are found.
2. For each ordered pair, or basis, of point features do the following:

(a) Compute the coordinates (u, v) of the remaining features in the coordinate frame defined by the basis.

(b) After proper quantization, use the tuple (u_q, v_q) as an index into a 2D hash table data structure and insert in the corresponding hash table bin the information $(m, (basis))$, namely the model number and the basis tuple used to determine (u_q, v_q).

Recognition phase

When presented with an input image, do the following:

1. Extract the various points of interest. Assume that S is the set of the interest points found; let *S* be the cardinality of S.
2. Choose an arbitrary ordered pair, or basis, of interest points in the image.
3. Compute the coordinates of the remaining interest points in the coordinate system *Oxy* that the selected basis defines.
4. Appropriately quantize each such coordinate and access the appropriate hash table bin; for every entry found there, cast a vote for the model and the basis.
5. Histogram *all* hash table entries that received one or more votes during step 4. Proceed to determine those entries that received more than a certain number, or threshold, of votes: Each such entry corresponds to a potential match.
6. For each potential match discovered in step 5, recover the transformation T that results in the best least-squares match between all corresponding feature pairs.
7. Transform the features of the model according to the recovered transformation T and verify them against the input image features. If the verification fails, go back to step 2 and repeat the procedure using a different image basis pair.

Figure 4. The two stages of the geometric hashing system [12]

To detect a planar marker in 3D space by geometric-hashing, it is required to do projective transformation between two planes (the given input image on camera screen and the marker image). We need a 4-point basis for such a projective transformation.

3. DESCRIPTION OF OUR ALGORITHM

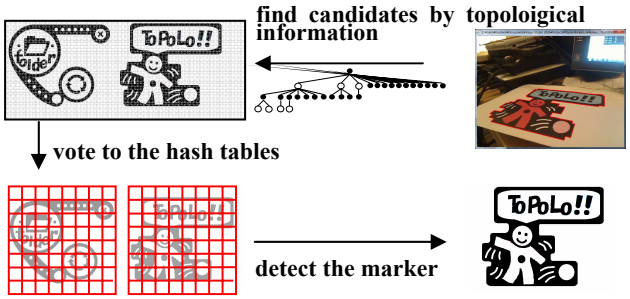

Figure 5. The overview of the phases in our method

Our marker tracking method combines two different approaches together. Figure 5 above describes the overview of our method. First, we use topological region adjacency to find marker candidates, using left heavy depth sequence. Then we apply a method similar to geometric-hashing, which determines the model by voting to the hash tables. Unlike traditional geometric-hashing, we prepare one hash table for each model and try all the possible combinations for 4-point basis to increase the robustness. Yet, as described in the later section, our method reduces the computational cost by use of topological information. The 4-point basis acquired in this phase can be used to estimate 6-DoF pose information.

3.1 Finding a Marker Candidate.

We use left heavy depth sequence as in *reacTIVision*. Figure 6 shows two examples of our markers, *folder* and *kid* (the colored versions are used in Figure 1). Notice these two markers share the same topological structure. The topological region adjacency tree of these two markers is shown in Figure 7 in the next section.

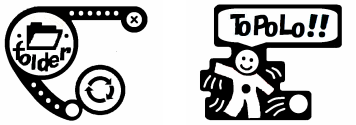

Figure 6. Two Examples of Our Markers

3.2 Determining a Model and 6-DoF Pose

Now we have the marker candidates found by left heavy depth sequence, yet the information obtained so far is not enough to distinguish the models with the same topological structure.

To determine a model, first we acquire the centroids of the regions of the leaf nodes in the topological region adjacency tree. Centroids are not projection-invariant, yet if the shape of region is simple and small, the error between actual centroid and the centroid in the projected image can be acceptable for the later phase to votes to the hash tables in our method.

One of the problem in geometric-hashing is that if the number of the points that can be picked up for basis is large, both computational cost and memory usage can be considerably expensive, especially when 4-point basis is required to find a homography as in this case. As the number of the possible combinations of the basis increases, the entries to hash table bins can significantly increase and may also result in more repetition of the vote-verify cycle until it acquires the satisfactory answer in the verification phase.

Our method reduces such a computational cost by grouping the leaf nodes by topological information. For instance, Figure 7 (above) shows the topological region adjacency tree and its left heavy depth sequence of the markers in Figure 6. The leaf nodes in Figure 7 can be divided into Group A to Group F as shown, by the routes to reach each group from the root.

To reach the leaf nodes in Group A, the route to take is made of the nodes noted as (i) and (ii). Taking the partial tree structures using these two nodes as the roots, the sequence of such partial tree structures can be described as in Figure 7 (middle). Using the parts from the original left heavy depth sequence, which are corresponding to each tree structure, the route can be described as 0123323232222122212111111111111→123323232222→233→3. Thus, those leaf nodes in the same group shares such the same route and the routes can be distinguished by such a sequence of left heavy depth sequences.

By considering such grouping when picking up the 4 points for basis, the number of the possible combinations for 4-point basis can be significantly reduced. In the case of the topological structure in Figure 7 (above) with 23 leaf nodes, without such a consideration the total number of the possible combinations for 4-point basis is 23*22*21*20=212,520. Yet, if we decide to pick up 2 points from Group B, 1 point from Group E and 1 point from Group F for basis, the number of the combinations can be reduced to 2*1*1*11=22. Thus, by taking topological information into account, our method significantly reduces the number of the possible combinations for basis.

Our method considers such grouping by topological information also in the phase to vote to the hash table, to improve the accuracy of the votes. Even when a vote to the hash table is casted to a bin where a point in a marker model is registered, the vote will not be treated as valid if the group of voting point and that of the registered point is not the same.

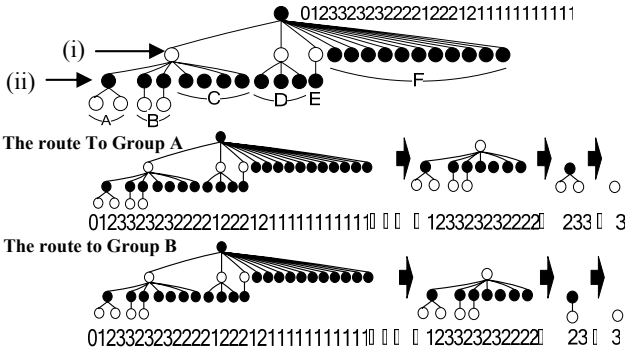

Figure 7. An example of a topological region adjacency tree (above) and the routes to Group A and Group B (below)

Then, as in geometric-hashing method, we will vote to the hash table where a marker model is registered. While traditional geometric-hashing repeatedly picks up the arbitrary basis until the evaluation result passes the verification, we try all the possible combinations of basis for each model. The most voted combination of the marker and the 4-point basis will be treated as a detected marker and the basis.

Since our method first selects the marker candidates by its topological structure and also reduces the possible combination for the basis by topological information as above, it can still run in real-time even when all the combinations of basis and models

are examined, as described in the later section. By preparing a hash table for each marker and testing all the possible combinations, our method can robustly detect the markers.

After the marker is determined in this phase, the matching of 4 points used for basis between input image and marker image can be acquired. These points can be used for 6-DoF pose estimation.

4. EXPERIMENTAL APPLICATION AND DISCUSSION

We have implemented a prototype system based on our method. We tested our prototype system on Viliv S5, a mobile Internet device that runs Windows XP, with Atom Z520/1.33GHz CPU and 1GB memory. Logitec QCam for Notebooks webcam is used as video input device. The pictures of the device are shown in Figure 8 below. We registered 1 marker, *kid* shown in Figure 6 and took the average processing time of 1000 frames. The average time cost for each frame was 74.6msec/frame, in 640x480 pixels (42.0msec for smoothing input image, 17.2msec for binarization, 10.8msec for topological information extracting, 4.6msec for marker detection by voting). Smoothing phase may be removed, depending on the input image quality. As this result shows, the system can run in real-time even on mobile Internet device.

As the previous research suggests, the topology-based method can be significantly robust against false detection, using markers with complex enough topological structures [2, 3]. Such robustness of topology-based method is also kept in our method and the prototype showed good robustness against false detection.

Figure 8. The picture of Viliv S5 and Nintendo DS (left) and a user detecting the markers with Viliv S5 (right)

We have also implemented a traditional geometric-hashing version for the prototype system; however, the result was less robust. Even though it is still possible to apply some additional techniques to improve the accuracy of geometric-hashing, our current method can try all the possible 4-point bases in acceptable computational cost, since our method first filters out the irrelevant markers with differing topological structure and then significantly reduces the possible combinations for 4-point basis by utilizing topological information. Such a reduction of the computational cost by topological information before the voting phase helps avoiding the damage to the over-all performance that can be caused by registering more markers to the system. Unlike our topology-based method, the pattern-matching method found in ARToolkit[7] is required to compare a marker candidate to all the registered markers by pattern-matching, which may significantly affect the over-all performance when many markers are registered.

Another issue to be considered would be the accuracy of 6-DoF pose estimation. Our method currently uses each centroid of 4-points used as basis to estimate 6-DoF pose. Since centroids are not projection-invariant, even though the errors between the actual centroids in marker coordinates and the projected image on camera screen can be acceptable for voting phase, these errors can cause less accurate 6-DoF pose estimation. However, it is still good enough for casual mobile interaction purposes, which rarely require very precise pose estimation.

5. CONCLUSION

We have developed a novel marker tracking technique wit shape-free, designable markers, with a focus on visually communicative design in mobile/pervasive environment. Our method is based on the combination of the topology-based approach and geometric hashing-like approach and is fast enough to run in real-time even on low performance CPUs found in mobile Internet devices. Our method can distinguish those markers with the same topological structure and is capable of 6-DoF pose estimation. Lack of such features has been a significant deficit in the most of the existing systems based only on topological region adjacency.

6. FUTURE WORK

We are planning to port the prototype systems to mobile phone devices, and prototype several camera-based interactive applications. We expect our method to run in real-time on the recent hardware such as Apple iPhone or Google Nexus One.

7. ACKNOWLEDGMENTS

The author thanks Information-Technology Promotion Agency Japan for supporting this software development through their Exploratory Software Project Fund.

8. REFERENCES

[1] Bay, H et al. 2008. SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110 (3), 346--359

[2] Bencina, R. et al. 2005. Improved Topological Fiducial Tracking in the reacTIVision system. Proc of CVPR'05

[3] Costanza, E. and Huang, J. 2009. Designable Visual Markers, Proc. Of CHI'09, 1879-1888

[4] Costanza, E. and Lennis, M. 2006. Telling a Story on a Tag: The Importance of Marker's Visual Design for Real World Applications. Proc. Of MIRW'06

[5] Fiala, M. 2005. ARTag, a Fiduial Marker System Using Digital Techiniques, Proc of CVPR'05, 590-596

[6] ISO. 2000. International Standard, ios/iec18004. ISO International Standard

[7] Kato, H. and Billinghurst, M. 1999, Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conference System. Proc of ISMAR'99

[8] Lowe, D. 2004. Distinctive image features from scale-invariant keypoints**.** Proc. Of Intl. Journal of Computer Vision**,** 60**(**2**),** 91-110.

[9] Nakano, S. and Uno, T. 2003. Efficient generation of rooted trees, National Institute for Informatics (Japan), Tech. Rep NII-2003-005E

[10] Rekimoto, J., Ayatsuka, Y. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. Proc of DARE'00

[11] Wagner, D and Schmalstieg, D. 2007. ARToolkit Plus for Pose Tracking on Mobile Devices. Proc. Of CVWW'07

[12] Woflson, J. and Rigoutsos, I. 1997. Geometric Hashing: An Overview, IEEE Computational Science & Engineering, Vol. 4(4). 10-21